|  |  |  |
| --- | --- | --- |
| C:\Users\amanu\AppData\Local\Microsoft\Windows\INetCache\Content.Word\logo iim.png | **Python**  **A3DW**  **TD 6 :** Fonctions, Boucles, Contrôles | |
|  |  |

# Rappels

# Exercices

## Fonctions

Pour chaque énoncé veuillez définir la fonction correspondante

### Fonction 1 (15min)

Veuillez réaliser la fonction splitCodon(dna) qui prend en argument une chaîne de caractères représentant de l’ADN et **retourne** la liste des codons la composant.

*Exemple :*

dna = 'TATTCGCTATGGCAT’

print(splitCodon(dna))

*Affichage :*

['TAT', 'TCG', 'CTA', 'TGG', 'CAT']

*Définition :*

### Fonction 2 (15min)

Veuillez réaliser la fonction countNuc(dna) qui prend en argument une chaîne de caractères représentant de l’ADN et **retourne** un dictionnaire contenant le nombre d’occurrences de chaque nucléotide.

*Exemple :*

dna = 'TATTCGCTATGGCAT’

print(countNuc(dna))

*Affichage :*

{'A':3, 'T':6, 'C':3, 'G':3}

*Définition :*

### Fonction 3 (15min)

Veuillez réaliser la fonction average(l) qui prend en argument une liste de note et **affiche** la moyenne.

*Exemple :*

notes = [10.5, 4, 15.5, 17, 2]

average(notes)

*Affichage :*

9.8

*Définition :*

### Fonction 4 (15min)

Veuillez réaliser la fonction minimum(l) qui prend en argument une liste de note et **affiche** la note minimale.

*Exemple :*

notes = [10.5, 4, 15.5, 17, 2]

minimum(notes)

*Affichage :*

2

*Définition :*

### Fonction 5 (15min)

Veuillez réaliser la fonction isIdentity(mx) qui prend en argument matrice représentant une matrice carrée au sens mathématique mathématique et vérifie si cette dernière est une matrice identité ou non. Elle **retourne** True ou False en conséquence.

*Exemple :*

matrix\_1 = [[1,2,3],[4,5,6],[7,8,9]]

matrix\_2 = [[1,0,0,0],[0,1,0,0],[0,0,1,0],[0,0,0,1]]

print(isIdentity(matrix\_1))

print(isIdentity(matrix\_2))

*Affichage :*

False

True

*Définition :*

### Fonction 6 (15min)

Veuillez réaliser la fonction countMutations(dna\_1, dna\_2) qui prend en argument deux chaînes de caractères représentant de l’ADN et **retourne** une matrice comptant les mutations entre les séquences. Les séquences sont forcément de même taille.

*Exemple :*

seq\_1 = 'TATTCGCTATGGCAT'

seq\_2 = 'GTTACGCTATCGTAA'

print(countMutations(seq\_1, seq\_2)

*Affichage :*

[' ', 'A', 'T', 'C', 'G'], ['A', 0, 1, 0, 0], ['T', 2, 0, 0, 1], ['C', 0, 1, 0, 0], ['G', 0, 0, 1, 0]]

*Définition :*